REMARKS

The foregoing Amendment after Final and the following Remarks are submitted in response to the Final Office Action issued on August 10, 2005 in connection with the above-identified patent application, and are being filed within the three-month shortened statutory period set for a response by the Final Office Action.

Claims 15, 16, 20-27, 31, 32, and 36-43 are pending in the present application. Claims 17, 18, 33, and 34 have been canceled and the subject matter thereof has been added to independent claims 15, 25, 31, and 41.

Applicants respectfully submit that the Amendment after Final should be entered inasmuch as such Amendment is believed to place the application in condition for allowance and is not believed to require further searching. Applicants also respectfully submit that the added subject matter in claims 15, 25, 31, and 41 is not new matter inasmuch as such subject matter was disclosed in the application as originally filed at least in claims 17, 18, 33, and 34 and in the disclosure with regard to steps 1403 and 1417 of Fig. 14.

As was previously pointed out, the present invention is directed toward the problem that for a computing device such as a portable computing device to be trusted in the context of a rights management architecture, the portable device and the processor thereon must be of a type that substantially completely prevents a content thief from performing nefarious acts that would allow obtaining of content therein in an unencrypted form or decryption keys. Thus, according to the present invention, the processor is a secure processor and is constructed to run only authorized code, and is operated to maintain a strict cryptographic separation between applications that may be instantiated thereon.

The secure processor is operable in a normal mode and a preferred mode, where the security kernel can access a locally accessible CPU key only during the preferred mode. The security kernel employs the accessed CPU key during the preferred mode to instantiate and/or authenticate a secure application such as a rights management system, a banking / financial system, etc. on the portable device. The security kernel may automatically instantiate a particular secure application, may authenticate a secure application instantiated by another process, or may initially instantiate a secure chooser application that allows a user to select from one or more available secure applications on the portable device.

In any case, the accessed CPU key is typically a symmetric key that is employed by the security kernel to decrypt one or more encrypted security keys for the application instantiated. The CPU key is accessible only by the security kernel and only during the preferred mode, and is the key to unlocking or decrypting the secrets identified with each application, and therefore must be well-protected.

The Examiner has again rejected claims 15, 16, 20-27, 31, 32, and 36-43 under 35 USC § 102(e) as being anticipated by Vu et al. (U.S. Patent No. 6,557,104). Applicants respectfully traverse the § 102(e) rejection insofar as it may be applied to the claims as amended.

Independent claim 15 of the present application recites a method for a secure processor to instantiate and authenticate a secure application thereon by way of a security kernel. In the method, the secure processor enters a preferred mode where a security key of the processor is accessible and instantiates and runs a security kernel. Thereafter, the security kernel accesses the security key and applies same to decrypt at least one encrypted key for the application, stores the decrypted key(s) in a location where the application will expect the

key(s) to be found, and authenticates the application on the processor. The secure processor then enters a normal mode from the preferred mode after the security kernel authenticates the application, where the security key is not accessible. Thus, the security kernel allows the processor to be trusted to keep hidden the key(s) of the application. The security kernel employs the accessed security key during the preferred mode to authenticate / verify the application prior to instantiation thereof.

Independent claim 31 recites the subject matter of claim 15, albeit in the form of a computer-readable medium.

As was previously pointed out, the Vu reference discloses a secure processor where, during run-time, an application requiring access to a secure service invokes a security routine that in turn invokes a security mode by way of a high-level security interrupt which cannot be otherwise invoked. As best set forth at column 5, lines 24-41, once in the security mode, a security function is invoked to access secrets and data from an otherwise inaccessible storage location, and the security function performs appropriate security functionality based on such secrets and data, such as for example encryption and decryption, password validation, user authentication, etc.

However, and significantly, the Vu reference does not disclose entering the preferred and normal modes as recited in claims 15 and 31.

Independent claim 25 recites a method for a secure processor to instantiate one of a plurality of available secure applications thereon by way of a security kernel. In the method, a chooser value is set to a value corresponding to a chooser application upon power-up, and a preferred mode is entered upon a power-up CPU reset and instantiating the security

kernel. The security kernel determines that the chooser value corresponds to the chooser application and therefore authenticates and instantiates same.

After the chooser application is instantiated, a normal mode is entered and the chooser application presents the plurality of available applications for selection by a user. Upon receiving a selection of one of the presented applications to be instantiated, the chooser value is set to a value corresponding to the selected application. Thereafter, a CPU reset is executed and the preferred mode is entered, and the security kernel is instantiated. The security kernel then determines that the chooser value corresponds to the selected application and therefore authenticates and instantiates same. Normal mode is then entered after the selected application is instantiated and run. Thus, the security kernel allows the processor to be trusted to keep hidden a secret of the chooser application and a secret of the selected application.

Although the Examiner again argues that the Vu reference discloses the chooser application, the chooser value, and the use thereof as recited in claims 25 and 41, Applicants again must disagree, and in fact submit that the Vu reference is utterly silent about employing such items to securely choose and instantiate one of a plurality of applications on a secure processor in the manner recited in claims 25 and 41. In particular, the Vu reference does not at all disclose or even suggest switching between modes as recited to first load and then operate a chooser application, employ same to select a chooser value corresponding to a chosen application, and then load and operate a chosen application in the manner recited in claims 25 and 41.

Nevertheless, Applicants respectfully point out that claims 15, 25, 31, and 41 as amended also recite that the processor has a cache, and that the method further comprises

erasing data in the cache of the processor when entering preferred mode such that any data previously stored in the cache is not available to interfere with preferred mode operations, and also erasing data in the cache of the processor when entering normal mode such that any sensitive data in the cache from preferred mode operations is not available during normal mode from such cache.

Thus, and as may be appreciated, by erasing the cache when transitioning between preferred and normal modes sensitive data that could reside in the cache is flushed. As a result, any code in the nature of a Trojan Horse or the like is not available to run during preferred mode and interfere with the security kernel during such preferred mode. Likewise, any residue that might for example contain the CPU key in a decrypted form is not available to a nefarious party during normal mode.

Significantly, the Vu reference is entirely silent with regard to erasing data in the cache of the Vu processor, and in particular does not disclose erasing such data when entering preferred mode such that any data previously stored in the cache is not available to interfere with preferred mode operations, and also does not disclose erasing data in the cache of the processor when entering normal mode such that any sensitive data in the cache from preferred mode operations is not available during normal mode from such cache, all as required by claims 15, 25, 31, and 41. Although the Examiner points to column 5, lines 42-47 and 59-63 in an effort to show otherwise, Applicants respectfully submit that such passages do not in fact teach erasing any Vu cache in the manner recited in the claims. In particular, such passages are set forth below:

> The appropriate cryptographic information is provided to the application program at step 25. The entire processing has occurred in a secure mode and a secure memory area which are not visible to the applications previously running on the processor.

> At this point, SMM takes over and the entire operating system and its applications are put into a "sleep mode." The operations to calculate the response based on the cryptographic key and the challenge are then performed (steps 22, 23).

and as can be plainly seen are utterly devoid of any mention of a cache, let alone erasing such a cache when entering preferred or normal modes.

Further, Applicants respectfully submit that the Vu reference is entirely silent with regard to and in fact does not suggest or even appreciate that such a cache should be erased in the manner recited. In particular, such Vu reference does not appreciate that such a cache should be erased when transitioning to preferred mode so that any code in the nature of a Trojan Horse or the like is not available to run during preferred mode and interfere with the security kernel during such preferred mode, and also that such a cache should be erased when transitioning to normal mode so that any residue that might for example contain the CPU key in a decrypted form is not available to a nefarious party during normal mode.

Accordingly, Applicants respectfully submit that the Vu reference does not anticipate or even suggest the invention recited in claims 15, 25, 31, and 41 or any claims depending therefrom. Accordingly, and for all the aforementioned reasons, Applicants respectfully request reconsideration and withdrawal of the § 102(e) rejection.

In view of the foregoing discussion, Applicants respectfully submit that the present application, including claims 15, 16, 20-27, 31, 32, and 36-43, is in condition for allowance, and such action is respectfully requested.

Respectfully submitted,

Date: September 30, 2005

Steven H. Meyer
Registration No. 37,189

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439